EM-algorithm in Software Reliability Modeling

James Ledoux
INSA de Rennes & IRMAR
20 avenue des Buttes de Coësmes
CS 14315
35043 Rennes Cedex, France
Email: james.ledoux@insa-rennes.fr

Abstract— **The purpose of this paper is to give an overview on the use of the Expectation-Maximization (EM) algorithm in software reliability modeling. This algorithm is related to Maximum Likelihood Estimates (MLE) of parameters in a context of missing data. Different ways to implement this algorithm are highlighted for hidden Markov models in software reliability.**

I. Introduction

Most systems are now driven by software. Thus, it is well recognized that assessing the reliability of software applications is a major issue in reliability engineering, particularly in terms of cost. But predicting software reliability is not easy. Perhaps the major difficulty is that we are concerned primarily with design faults, which is a very different situation from that tackled by conventional hardware theory. A *fault* (or bug) refers to a manifestation in the code of a mistake made by the programmer or designer with respect to the specification of the software. Activation of a fault by an input value leads to an incorrect output. Detection of such an event corresponds to an occurrence of a software *failure*. Input values may be considered as arriving to the software randomly. So although software failure may not be generated stochastically, it may be detected in such a manner. Therefore, this justifies the use of stochastic models of the underlying random process that governs the software failures. Two approaches are used in software reliability modeling. The most prevalent is the so-called *black-box* approach, in which only the interactions of the software with the environment are considered. Following Gaudoin [1] and Singpurwalla and Wilson [2], the *self-exciting point processes* can be used as a basic tool to model the failure process. This enables an overview of most of the published Software Reliability Models (SRMs). We also refer to Musa *et al.* [3], the recent book by Pham [4] and the handbook [5] for a complete view. A second approach, called the *white-box* approach, incorporates information on the structure of the software in the models (see [6, and the references therein]). In fact, this approach generates a class of models that can be analyzed by martingale methods in the framework of point processes [7]. An archetype of this class of models will be discussed in Section IV.

Suppose that we have selected one model, the unknown parameters have to estimated from the failure data. These data may come from different phases of the life cycle of the software : testing, operational, …The MLE and the Bayesian estimation methods are the standard methods for calibrating black-box models. This is well documented (e.g. see [4], [2]). The practical implementation of these estimation procedures must be carefully performed. Indeed, roughly speaking, the use of MLE involves routines for solving strongly non-linear equations and the Bayesian estimates need Monte Carlo Markov Chain methods for computing multi-dimensional integrals. The architecture-based approach adds complexity to the models, the data collection and the statistical analysis as well. In general, some estimates may be obtained from data collected in the earlier phase of the software life cycle [6]. But little has be done on the statistical analysis of the architecture-based models.

Recently, the EM-algorithm has been considered for estimating the parameters of SRMs [8], [9], [10], [11], [12], [13], [14]. This is related to MLE in the context of missing data. Two basic point of view are mainly used.

- For a special but wide class of (discrete or continuous-time) "black-box" models, the basic parameters are estimated on the basis that the failure data gathered at a specific instant may be regarded as incomplete data [9], [10], [11].
- The SRM is directly based on a Hidden Markov Chain (HMC), or a partially observable Markov chain. Then, the parameters of the non-observable part of the model (that is a Markov chain), are estimated from the incomplete or observed data [12], [13], [14].

Some details will be reported in Section II on the first point of view. But, here, we are mainly concerned with HMCs. In the next section, we will introduce an abstract version of the EM-algorithm and it will be specialized to the HMCs. This is an iterative algorithm where the value of the parameters of the non-observed Markov chain are up-dated in regard of the observation of new data. This updating involves the computation of expectation of statistics of the non-observed Markov chain conditionally to the available data. The main purpose of the paper is to give an overview of three basic methods to implementing such a computation. We examine the case of finite discrete-time HMCs in Section III and the case of so-called Markovian Arrival Processes in Section IV. The basic way to implement the EM-algorithm is to use a Baum-Welch or "Forward-Backward" principle (see Subsection III-A). The derivation of so-called *smoothers* is required. In the discrete-time context, we mention a simple and direct way to replace the usual "forward-backward" smoothing by recursive smoothing. Finally, a filter-based approach as advocated in [15] is discussed.

II. EM-ALGORITHM

A. The algorithm

Suppose that some variables Y are observed, but that there exist additional variables X that we cannot observe. Let $L(\theta; Y)$ be the *observed data likelihood* of a parameter θ given the observations Y, and let $L^c(\theta; Y, X)$ be the *complete data likelihood* of the parameter, also including the missing data. If θ is a given parameter estimate, then it can be shown that the estimate

$$\widehat{\theta} = \arg\max_{\theta^*} Q(\theta^* \mid \theta) \tag{1}$$

where

$$Q(\theta^* \mid \theta) = \mathbb{E}_\theta\big[\log L^c(\theta^*; Y, X) \mid Y\big], \tag{2}$$

makes the observed data likelihood non-decreasing, i.e. $L(\widehat{\theta}; Y) \geq L(\theta; Y)$ [16]. Then, this procedure can be iterated. The evaluation of the conditional expectation in (2) is called the *E-step* of the algorithm, and the maximization in (1) is the *M-step*. In many cases, the likelihood $L(\theta; Y)$ is highly non-linear in θ and difficult to maximize, while the M-step of the EM-algorithm, involving the complete data likelihood L^c, is often explicit. This is the main reason for the widespread use of the EM-algorithm.

Example 1 ([11]): A class of discrete-time Non-Homogeneous-Poisson-Processes is considered. That is, we have a sequence random variables $(N_t)_{t\in\mathbb{N}}$ (with $N_0 := 0$) which has independent increments and any increment $N_{t+h} - N_t$ (with $h \geq 1$) has a Poisson distribution with parameter $\Lambda_{t+h} - \Lambda_t$ where $\Lambda_t = \mathbb{E}[N_t]$. Here, the expectation function is assumed to be $\Lambda_t = \omega F(t)$ with $F(t)$ is the distribution function of some parametric probability distribution on the positive integers. For instance, let us consider the discrete-time counterpart of the Goel-Okumoto model for which $F(t) := 1 - (1-b)^t$ for $t \geq 1$. The authors take the sequence of independent random variables $(X_t := N_t - N_{t-1})_{t\geq 1}$ as the complete data and $\theta = \{\omega, b\}$ as parameters to be estimated. Suppose that the observed data are $X_1, \ldots, X_t$. Then, it is easily seen that

$$\begin{aligned} Q(\theta^* \mid \theta) = K + \mathbb{E}[\sum_{l=1}^{\infty} X_l \mid X_1, \ldots, X_t] \log \omega^* - \omega^* \\ + \mathbb{E}[\sum_{l=1}^{\infty} X_l \log\big(1 - (1-b^*)^l\big) \mid X_1, \ldots, X_t] \end{aligned}$$

where K does not depend upon θ^*. Thanking the independence of the data, we can easily obtain the following re-estimation formula maximizing $Q(\theta^* \mid \theta)$ under the constraints that $\omega^*, b^* > 0$:

$$\widehat{\omega} = N_t + \omega(1-b)^t \quad \widehat{b} = \frac{N_t + \omega(1-b)^t}{\sum_{l=1}^t lX_l + \omega(1-b)^t(t + 1/b)}.$$

The EM-algorithm was first designed as an estimation method for HMCs (e.g [17]). Here, an HMC is a bivariate

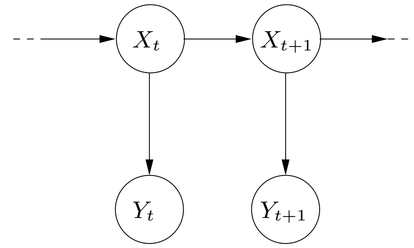

Fig. 1. Graphical representation of the dependence structure of an HMC

discrete-time homogeneous Markov chain $(Y_t, X_t)_{t\in\mathbb{N}}$ such that

- for any t, the conditional distribution of (Y_{t+1}, X_{t+1}) given (Y_t, X_t) does not depend on X_t and
- the conditional distribution of Y_{t+1} given X_{t+1}, X_t only depends on X_{t+1}.

Note the first property implies that $(X_t)_{t\in\mathbb{N}}$ is a homogeneous Markov chain. In general, the second property is called the *factorization property* of the transitions probabilities of $(Y_t, X_t)_{t\in\mathbb{N}}$. Both properties take the following form when $(Y_t, X_t)_{t\in\mathbb{N}}$ is finite-valued

$$\begin{aligned} &\mathbb{P}\{Y_{t+1} = f_k, X_{t+1} = e_j \mid Y_t = f_m, X_t = e_i\} \\ &= \mathbb{P}\{Y_{t+1} = f_k, X_{t+1} = e_j \mid X_t = e_i\} \\ &= \mathbb{P}\{Y_{t+1} = f_k \mid X_{t+1} = e_j\}\mathbb{P}\{X_{t+1} = e_j \mid X_t = e_i\}. \end{aligned} \tag{3}$$

The dependence structure of an HMC can be represented by a *graphical model* as in Figure 1. The dependence structure among random variables is described by a directed graph without loops. An equivalent and standard definition of an HMC in statistical literature is the following. This is a bivariate discrete-time process $(Y_t, X_t)_{t\in\mathbb{N}}$ such that

- $(X_t)_{t\in\mathbb{N}}$ is a Markov chain and
- $(Y_t)_{t\in\mathbb{N}}$ is a sequence of independent random variables conditionally on $(X_t)_{t\in\mathbb{N}}$ and, for any $t \in \mathbb{N}$, the conditional distribution of Y_t only depends on X_t.

The component $(Y_t)_{t\in\mathbb{N}}$ is supposed to be observable and the Markov chain $(X_t)_{t\in\mathbb{N}}$ is non-observed or hidden. The question is to estimate some unknown parameters of the HMC from the observations $(Y_t)_{t\in\mathbb{N}}$. This includes the transition probability matrix of the Markov chain $(X_t)_{t\in\mathbb{N}}$, the probability distribution of X_0 and parameters of the conditional distribution of Y_t on X_t.

Example 2 ([12]): An SRM is defined as follows. The sequence of failure rates of a software is supposed to form a finite Markov chain $(X_t)_{t\geq 1}$ with the state space $\{e_1, \ldots, e_n\}$. The time between failures $(Y_t)_{t\geq 1}$ are assumed to be independent conditionally to $(X_t)_{t\geq 1}$. Finally, conditionally to $\{X_t = e_i\}$, Y_t has an exponential distribution with parameter e_i. Thus, $(Y_t, X_t)_{t\geq 1}$ is an HMC with transition kernel

$$\begin{aligned} &\mathbb{P}\{Y_{t+1} \in \mathrm{d}s, X_{t+1} = e_j \mid X_t = e_i\} \\ &\qquad = \mathbb{P}\{Y_{t+1} \in \mathrm{d}s \mid X_{t+1} = e_i\}\mathbb{P}\{X_{t+1} = e_j \mid X_t = e_i\} \\ &\qquad = e_i \exp(-e_i s)\mathrm{d}s \times \mathbb{P}\{X_{t+1} = e_j \mid X_t = e_i\}. \end{aligned}$$

We point out that the number of hidden states n is a priori unknown. This is a major issue in hidden Markov modeling (e.g. see [18] for a recent discussion). The main purpose of [12] is to identify the major updates of the software – or to determine homogeneous periods in the debugging process. This leads to the well-known problem of restoration of the hidden states. We refer to [12] for details. A somewhat similar model was considered in [8], but in a quite different purpose.

III. The discrete state space case

In this section, we present different ways to implement EM algorithm for a Markov chain $(Y,X) := (Y_t,X_t)_{t\in\mathbb{N}}$. The state space of (Y,X) is assumed to be $\mathscr{Y}\times\mathscr{X}$, where $\mathscr{X} := \{e_1,\dots,e_n\}$, $\mathscr{Y} := \{f_1,\dots,f_n\}$, and its transition probabilities satisfy: for all $t\in\mathbb{N}$, $i,j=1,\dots,n$, $k=1,\dots,m$ (see (3))

$$\begin{aligned}&\mathbb{P}\{Y_{t+1}=f_k,X_{t+1}=e_j \mid Y_t=f_l,X_t=e_i\}\\ &=\mathbb{P}\{Y_{t+1}=f_k,X_{t+1}=e_j \mid X_t=e_i\}=P(i,j)\,G(j,k).\end{aligned}$$

where P is the $n\times n$ transition matrix of the Markov chain $(X_t)_{t\in\mathbb{N}}$ and G is a $n\times m$ stochastic matrix with the probability distribution $G(j,\cdot)$ of Y_t given $X_t=e_j$ as jth row. The parameter vector θ encompasses the entries of P as well as those of G. For a fixed parameter vector θ, we denote the underlying probability measure and associated expectation respectively by $\mathbb{P}_\theta$ and $\mathbb{E}_\theta$. X_0 or its probability distribution x_0 is assumed to be known.

The log-likelihood function for the complete data up to time t under $\mathbb{P}_\theta$ is denoted by $L^c_t(\theta;Y,X)$ and is defined as

$$\begin{aligned}\log L^c_t(\theta;Y,X) \;&:=\; K+\sum_{l=1}^{t}\log P(X_{l-1},X_l)\\ &\quad+\sum_{l=0}^{t}\log G(X_l,Y_l).\end{aligned}\tag{4}$$

where K is a constant that does not depend on θ. The formulas for reestimating θ from the observations $\mathbb{F}^Y_t := \sigma(Y_0,\dots,Y_t)$, are obtained using EM:

1) Initialization : choose $\theta^{(0)}$
2) *E-step.* Set $\theta := \theta^{(m)}$ and compute $Q(\cdot\mid\theta)$ in (2)

$$\begin{aligned}Q(\theta^*\mid\theta) &= \mathbb{E}_\theta\big[\log L^c_t(\theta^*;Y,X)\mid \mathbb{F}^Y_t\big]\\ &= \sum_{i,j=1}^{n}\log P^*(i,j)\;\widehat{N}^{ij}_t\\ &\quad+\sum_{k=1}^{m}\sum_{j=1}^{n}\log G^*(j,k)\;\widehat{G}^{jk}_t\end{aligned}\tag{5}$$

where

$$\widehat{N}^{ij}_t := \mathbb{E}_\theta[\sum_{l=1}^{t}1_{\{X_{l-1}=e_i,X_l=e_j\}}\mid\mathbb{F}^Y_t]\tag{6}$$

$$\widehat{G}^{jk}_t := \mathbb{E}_\theta[\sum_{l=0}^{t}1_{\{X_l=e_j,Y_l=f_k\}}\mid\mathbb{F}^Y_t]\tag{7}$$

are the expectations, given the observations, of the number of jumps from state e_i to e_j for X up to time t and of the number of visits to the joint state (e_j,f_k) for (Y,X) up to time t, respectively.
3) *M-step.* Determine $\theta^{(m+1)}$ maximizing the function (5) under constraints $\sum_{j=1}^n P^*(i,j)=1,\sum_{k=1}^m G^*(j,k)=1$ with $i,j=1,\dots,n$. Then, the following re-estimation formulas at step m are obtained using the Lagrange multipliers method

$$\widehat{P}^{(m+1)}(i,j)=\frac{\widehat{N}^{ij}_t}{\widehat{\mathcal{O}^{(i)}}_t},\quad \widehat{G}^{(m+1)}(j,k)=\frac{\widehat{G}^{jk}_t}{\widehat{\mathcal{O}^{(i)}}_{t+1}}\tag{8}$$

where

$$\widehat{\mathcal{O}^{(i)}}_t := \mathbb{E}_\theta[\sum_{l=0}^{t-1}1_{\{X_l=e_i\}}\mid\mathbb{F}^Y_t]\tag{9}$$

is the conditional expectation of the number of visits of X to state e_i up to time $t-1$ given the observation.
4) Return in 2 until a stopping criterion is satisfied.

An intuitive support for formulas (8) is that they involve the conditional expectations to the observations of the estimators obtained in maximizing the complete data likelihood, that is using standard MLE.

The last step is to evaluate the formulas in (8). This can be carried out using different ways. The conditional expectations (6,7,9) may be thought of as

- either functionals of the conditional expectations of the basic statistics of the HMC

$$1_{\{X_l=e_i\}},\; 1_{\{X_l=e_j,Y_l=f_k\}},\; 1_{\{X_{l-1}=e_i,X_l=e_j\}}$$

- or conditional expectations of the additive functionals of Markov chains

$$N^{ij}_t := \sum_{l=1}^{t}1_{\{X_{l-1}=e_i,X_l=e_j\}},\tag{10}$$

$$\mathcal{O}^{(i)}_t := \sum_{l=0}^{t-1}1_{\{X_l=e_i\}}\;\; G^{jk}_t := \sum_{l=0}^{t}1_{\{X_l=e_j,Y_l=f_k\}}.\tag{11}$$

The computational task associated with the first point of view is discussed in the next two subsections as well as for the second one in the third subsection. We refer to the delightful monograph [18] for a more complete discussion of these issues.

A. Smoothing by the Forward-Backward principle

We briefly present the so-called "forward-backward" or Baum-Welch strategy for the computation of the *smoothed state probabilities*: $l=0,\dots,t-1$ and $i=1,\dots,n$

$$\mathbb{P}\{X_l=e_i\mid\mathbb{F}^Y_t\}.\tag{12}$$

The conditional probability

$$\mathbb{P}\{X_t=e_i\mid\mathbb{F}^Y_t\}$$

is denoted by $\widehat{X}_t(i)$ and the vector $\widehat{X}_t$ is called the *state filter* at time t. In the sequel, for any $k=1,\dots,m$, the $n\times n$

diagonal matrix with entries $G(i,k), i=1,\ldots,n$ will be denoted by $\operatorname{diag}\big(G(\cdot,k)\big)$. With a slight abuse of notation, $\operatorname{diag}\big(G(\cdot,Y_t)\big)$ is $\sum_{k=1}^m \operatorname{diag}\big(G(\cdot,k)\big)1_{Y_t=f_k}$. The smoothed state probabilities are derived as: for $l=0,\ldots,t-1$

$$\mathbb{P}\{X_l=e_i \mid \mathbb{F}_t^Y\} = \frac{\widehat{X}_l(i)\beta_{l|t}(i)}{\sum_{i=1}^n \widehat{X}_l(i)\beta_{l|t}(i)}.$$

where the vectors $\beta_{k|n}$ and $\widehat{X}_l$ are computed from the well-known recursive two-pass through the data [19]:
Forward filtering.

$c_0 := x_0\mathrm{diag}(G(\cdot,Y_0))\mathbf{1}^\top$ and $\widehat{X}_0 := x_0\mathrm{diag}(G(\cdot,Y_0))/c_0$
for $l=1,\ldots,t:$ $c_l := \widehat{X}_{l-1}P\mathrm{diag}(G(\cdot,Y_l))\mathbf{1}^\top$
$\widehat{X}_l := \widehat{X}_{l-1}P\mathrm{diag}(G(\cdot,Y_l))/c_l$

Backward smoothing.

$\beta_{t|t} := \mathbf{1}/c_t$
for $l=t-1,\ldots,0$: $\beta_{l|t} := \beta_{l+1|t}\mathrm{diag}(G(\cdot,Y_l))P^\top/c_l$

For each run of EM, the computational cost is linear in the number of observations t and quadratic in the number of hidden states. The storage cost is linear in the number t of observations.

It is easily seen that, for $l=1,\ldots,t$,

$$\mathbb{P}\{X_{l-1}=e_i, X_l=e_j \mid \mathbb{F}_t^Y\} = \widehat{X}_{l-1}(i)P(i,j)G(j,Y_l)\beta_{l|t}(j).$$

Finally, we obviously have for all $l=1,\ldots,n$,

$$\mathbb{P}\{X_l=e_j, Y_l=f_k \mid \mathbb{F}_t^Y\} = 1_{\{Y_l=f_k\}}\mathbb{P}\{X_l=e_j \mid \mathbb{F}_t^Y\}.$$

When usual continuous-range observations are considered, the forward-backward strategy takes a very similar form. Specifically, if each conditional distribution of Y_t given $X_t = e_i$ has a probability density function $G(i,\cdot)$ with respect to the Lebesgue measure, we just have to replace in (4) the discrete kernel $(G(i,k))_{k=1}^m$ by the density function $G(i,\cdot)$. For instance, in Durand and Gaudoin's model (Example 2), the density function $G(i,\cdot)$ is an exponential density with parameter e_i. In such a case the parameter vector θ encompasses the entries of the transition probability P and the m parameters $\{e_1,\ldots,e_n\}$ of the family of exponential distributions (for sake of briefness, we assume that the distribution of X_1 is known). The log-likelihood in (4) has the form

$$\log L_t^c(\theta;Y,X) := K + \sum_{l=2}^t \log P(X_{l-1},X_l) + \sum_{l=1}^t \log X_l - \sum_{l=1}^t X_l Y_l,$$

where K is a constant that does not depend on θ. Then, using the EM strategy, we find the same formula as in (8) for the transition probabilities and the following formula for the parameters of the exponential distributions

$$\widehat{e}_i^{(m+1)} = \left(\frac{\sum_{l=1}^t Y_l\ \mathbb{E}_\theta[1_{\{X_l=e_i\}} \mid \mathbb{F}_t^Y]}{\mathbb{E}_\theta[\sum_{l=1}^t 1_{\{X_l=e_i\}} \mid \mathbb{F}_t^Y]}\right)^{-1}$$

This re-estimation formula can be implemented in computing the smoothed probabilities by the forward-backward principle. The only change in the algorithm is that the discrete probability distributions $G(i,\cdot), i=1,\ldots,n$ are replaced by the probability densities.

B. Recursive smoothing

We know from the forward filtering above, that the state filter $\widehat{X}_t$ may be computed from a formula which is recursive in the number of observations t. It turns out that the smoothed state probabilities may also be recursively computed, with a recursion in the number of observations. This fact has been pioneered by Elliott for HMCs but is not as known as the forward-backward principle (e.g. see [15]). The basic tool in [15] for deriving such recursive smoothing is the change of measure technique. For discrete state space HMCs, this technique may be replaced by a simple conditional expectation computation [20]. Let us derive now a recursive formula for the smoothed probabilities in (12). Then, the corresponding result for the smoothed joint probability distribution of (X_{l-1},X_l) is

$$\widehat{X}_{l-1}(i)P(i,j)G(j,Y_l)/c_l.$$

Lemma 1 (Recursive smoother for the state): The notation are as in the description of the forward-backward algorithm. We have for $l < t+1$, $\mathbb{P}\{X_l=e_i \mid \mathbb{F}_{t+1}^Y\} = \sum_{j=1}^n \gamma_{l,t+1|t+1}(i,j)$ where $\gamma_{l,t+1|t+1}(i,j) := \mathbb{P}\{X_l = e_i, X_{t+1}=e_j \mid \mathbb{F}_{t+1}^Y\}$). The matrix $\gamma_{l,t+1|t+1}$ satisfies the following recursive equation

$$\gamma_{l,t+1|t+1} \quad = \quad \gamma_{l,t|t}P\mathrm{diag}(G(\cdot,Y_{t+1})/c_{t+1}.$$

with $\gamma_{t,t|t} := \mathrm{diag}(\widehat{X}_t)$ for $t \geq 0$.

Proof: We know from [20, Lemma 5.1] that $\mathbb{P}\{X_l = e_i, X_{t+1}=e_j \mid \mathbb{F}_{t+1}^Y\}$ may be formulated as follows:

$$\begin{aligned} &\mathbb{P}\{X_l=e_i, X_{t+1}=e_j \mid \mathbb{F}_{t+1}^Y\} \\ = \ &\sum_{k=1}^m \frac{\mathbb{P}\{X_l=e_i, X_{t+1}=e_j, Y_{t+1}=f_k \mid \mathbb{F}_t^Y\}}{\mathbb{P}\{Y_{t+1}=f_k \mid \mathbb{F}_t^Y\}} 1_{\{Y_{t+1}=f_k\}}. \end{aligned}$$

Next, let us consider the numerator of the fraction above:

$$\begin{aligned} &\mathbb{P}\{X_l=e_i, X_{t+1}=e_j, Y_{t+1}=f_k \mid \mathbb{F}_t^Y\} \\ = \ &\mathbb{E}[1_{\{X_l=e_i\}}\mathbb{E}[\mathbb{P}\{X_{t+1}=e_j, Y_{t+1}=f_k \mid \mathbb{F}_t^{Y,X}] \mid \mathbb{F}_t^Y] \\ = \ &\mathbb{E}[1_{\{X_l=e_i\}}\sum_{p=1}^n \mathbb{E}[1_{\{X_t=e_p\}}P(p,j)G(j,k)] \mid \mathbb{F}_t^Y] \\ &\text{by the Markov property of } (Y_t,X_t)_{t\in\mathbb{N}} \\ = \ &\sum_{p=1}^n \mathbb{P}\{X_l=e_i, X_t=e_p \mid \mathbb{F}_t^Y\}P(p,j)G(j,k) \\ = \ &\sum_{p=1}^n \gamma_{l,t|t}(i,p)P(p,j)G(j,k). \end{aligned}$$

The probability $\mathbb{P}\{Y_{t+1} = f_k \mid \mathbb{F}_t^Y\}$ is the summation of the term above over i and j. Thus, we obtain

$$\begin{aligned}&\sum_{p=1}^{n} \mathbb{P}\{X_t = e_p \mid \mathbb{F}_t^Y\} \sum_{j=1}^{n} P(p,j)G(j,k) \\ =\ & \widehat{X}_t P \mathrm{diag}(G(\cdot,k))\mathbf{1}^\top = c_{t+1}\ 1_{\{Y_{t+1}=f_k\}}.\end{aligned}$$

■

The main advantage of this forward smoothing is to give an algorithm for which the data may be processed as they are collected. The number of observations has not to be fixed for computing the state smoothers. The computational complexity is cubic in the number of hidden states and linear in the number of observations. But the amount of storage does not depend of the number t of observations. Such a recursive smoothing may be used in Example 2 since the re-estimation formulas for its parameters only involve the computation of smoothers.

C. Filter-based approach

Now, we consider the expectations in (6,7,9) as conditional expectations of the additive functionals in (10-11). Using the filter-based approach pioneering by Elliott (e.g. see [15]), a recursive form of these conditional expectations may be obtained. Note that the technique of measure change used in [15] is not needed to obtain such recursive forms. Here, a direct proof using [20, Lemma 5.1] as for Lemma 1 could be given. In fact, the same trick than for the recursive computation of the smoothed probabilities is used. Recursive equations for the "joint statistics" of the functional under consideration with the hidden state are derived. The conditional expectations (6,7,9) are deduced from a "marginal distribution" computation.

Lemma 2: The meaning of the constant c_0 is provided in the description of the forward-backward principle. The state indicators vector $(1_{\{X_t=e_i\}})_{i=1}^n$ is denoted by p_t and $\boldsymbol{e}_i, i = 1,\ldots,n$ is the ith vector of the canonical basis of $\mathbb{R}^n$. The matrix $P\mathrm{diag}\big(G(\cdot, Y_{t+1})\big)$ is denoted $D_{Y_{t+1}}$ is the equations below. Let $\widehat{Zp}_t$ be the conditional expectation $\mathbb{E}[Z_t p_t \mid \mathbb{F}_t^Y]$ provided that the expectation is well defined. Note that $\mathbb{E}[Z_t \mid \mathbb{F}_t^Y] = \langle \widehat{Zp}_t, \mathbf{1}\rangle$.

1) *Number of visits to a hidden state.* We have $\widehat{\mathcal{O}^{(i)}p}_0 = (x_0(i)G(i,Y_0)/c_0)\,\boldsymbol{e}_i$ and for $t \geq 0$

$$\widehat{\mathcal{O}^{(i)}p}_{t+1} = \frac{\widehat{\mathcal{O}^{(i)}p}_t\, D_{Y_{t+1}} + \widehat{X}_t(i)\,\boldsymbol{e}_i\, D_{Y_{t+1}}}{\widehat{X}_t D_{Y_{t+1}}\mathbf{1}^\top}.$$

2) *Number of jumps of the hidden Markov chain.* We have $\widehat{N^{ij}p}_0 = 0$ and for $t \geq 0$

$$\widehat{N^{ij}p}_{t+1} = \frac{\widehat{N^{ij}p}_t D_{Y_{t+1}} + D_{Y_{t+1}}(i,j)\widehat{X}_t(i)\ \boldsymbol{e}_j}{\widehat{X}_t D_{Y_{t+1}}\mathbf{1}^\top}.$$

3) *Number of visits to a joint state.* $\widehat{G^{jk}p}_0 = 1_{\{Y_0=f_k\}}\big(x_0(i)G(i,k)/\sum_i x_0(i)G(i,k)\big)\boldsymbol{e}_i$ and for $t \geq 0$

$$\widehat{G^{jk}p}_{t+1} = \frac{\widehat{G^{jk}p}_t D_{Y_{t+1}}}{\widehat{X}_t D_{Y_{t+1}}\mathbf{1}^\top} + \frac{(\widehat{X}_t D_k)(j)}{\widehat{X}_t D_k \mathbf{1}^\top} 1_{\{Y_{t+1}=k\}}.$$

The main feature of the filter-based approach is to solve a recursion for *each* additive functional required by the EM strategy. In contrast, the forward-backward approach only need the computation of the smoothed probabilities. At each run of EM, the computational cost is linear in the number of observations t and of order 4 in the number of parameters (due to the n^2 statistics N_t^{ij}, $i,j = 1,\ldots,n$). This method is not very competitive from the computational point of view. But it presents the advantage to be recursive in the number of observations, that is, only the present estimates have to be stored. The storage cost does not depend on the number of observations which is interesting when the data sets are large.

IV. Architecture-based software reliability modeling

In this section, we consider some architecture-based software reliability models or "white-box" models for which parameter estimation may be carried out by the EM-algorithm.

A. A discrete-time architecture-based model

We briefly describe an architecture-based software reliability model, which can be viewed as an elaboration of Cheung's model [21]. Many aspects of the model appear to be limitations. Some of them can be overcame from [22], [23]. We are concerned with a model that uses the control graph to represent the architecture of the system. The transfers of control between modules are assumed to have Markov dynamics. Therefore, the execution model of the software is a discrete-time Markov chain $(C_t)_{t\in\mathbb{N}}$ on the state space $\mathscr{X} = \{e_1,\ldots,e_n\}$, where $\mathscr{X}$ may be thought of as the set of modules. This Markov chain is specified by its transition matrix A and by the probability distribution of C_0.

Let us describe the failure process. A first type of failure is associated with the visits to states, a second one with the transitions between states. When the model is in state e_i, a failure occurs with probability p_i. For simplicity, the time to recover a safe state is neglected (see [22]). Then state e_j is entered with constant probability $\alpha(i,j)$ (with $\sum_{j=1}^n \alpha(i,j) = 1$). In some applications, it can be useful to associate failure events directly with transitions. To do this, suppose that a failure does not occur during a visit to state e_i (this event has probability $1-p_i$). If the next state to be visited is state e_j (that happens with probability $A(i,j)$), a *transfer failure* may happen with probability $\lambda_{i,j}$. Then state e_l is entered with constant probability $\alpha^{i,j}(i,l)$ (with $\sum_{l=1}^n \alpha^{i,j}(i,l) = 1$).

Let us define the process $X := (X_t)_{t\in\mathbb{N}}$ where X_t is the occupied state at t. We define $n \times n$ matrices D_0 and D_1 by

$$\begin{aligned}D_0(i,j) &:= (1-p_i)A(i,j)(1-\lambda_{i,j})\\ D_1(i,j) &:= p_i\,\alpha(i,j) + \big[\textstyle\sum_{l=1}^n (1-p_i)\,\lambda_{i,l}\,A(i,l)\big]\alpha^{i,l}(i,j)\end{aligned} \tag{13}$$

Then X is a Markov chain with transition matrix $P = D_0 + D_1$. The entry $D_0(i,j)$ – $D_1(i,j)$ – represents the probability that X jumps from state e_i to e_j with no – one – failure event.

Let us consider the process $(N,X) := (N_t, X_t)_{t\in\mathbb{N}}$ over the state space $\mathbb{N}\times\mathscr{X}$, where $N := (N_t)_{t\in\mathbb{N}}$ is the counting process of failures. It follows from the various assumptions that (N,X) is a Markov chain with transition probabilities satisfying for all $m \in \mathbb{N}$, $t \geq 1$, $i,j = 1,\dots,n$, $k = 0,1$

$$\begin{aligned}&\mathbb{P}\{N_t = m+k, X_t = e_j \mid N_{t-1} = m, X_{t-1} = e_i\}\\ =&\mathbb{P}\{N_t - N_{t-1} = k, X_t = e_j \mid X_{t-1} = e_i\} = D_k(i,j).\end{aligned} \tag{14}$$

The other transition probabilities are zero. The transition matrix has the following special structure when the states are listed in lexicographic order

$$A = \begin{pmatrix} D_0 & D_1 & \mathbf{0} & \cdots \\ \mathbf{0} & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{pmatrix}. \tag{15}$$

The failure process associated with such models turns to be a particular instance of a class of discrete-time point processes known as the Markovian Arrival Processes (MAP) (e.g. see [24]). The distribution of the cumulative number of failures up to time t, N_t, is computed using a system of difference equations that only involve the matrices D_0, D_1. Thus, the knowledge of the non-negative parameter vector

$$\theta = \{D_k(i,j), \quad k = 0,1 \quad i,j = 1,\dots,n\}$$

satisfying $\sum_{k=0}^{1}\sum_{j=1}^{n} D_k(i,j) = 1$ for every i, is needed.

The major problem is to estimate all the parameters in (13). In general, we can obtain a priori estimates for the parameters of the model using procedures reported in [6]. They are based on data collected at earlier phases of the software life cycle (validation phases, integration tests,...). Sometimes, these estimates might appear to be rough estimates when the software is in operation. Here, the EM-algorithm provides a method for updating the estimates of matrices D_k during the life of the system. The only available data are the failure events. In that perspective, the Markov chain (N,X) should be thought of as a partially observed Markov model.

The counting variable N_t is the sum of the Bernoulli random variables $\Delta N_m = N_m - N_{m-1}$ for $m \geq 1$ and $\Delta N_0 = 0$. We know from (14) that the bivariate process $(\Delta N, X)$ is a Markov chain with finite state space $\{0,1\} \times \mathscr{X}$. Its transition probabilities are given in (14). We note that $(\Delta N, X)$ is not a standard Hidden Markov chain, in the sense that its transition probabilities do not satisfy the factorization property (3). However, it can be seen that this special property has no influence on the development of an EM-algorithm as in the previous section. In other words, there is an EM-algorithm framework for HMCs and for the partially observable Markov chains as well. We do not give the full details. It is quite similar to the discussion for the HMCs, once the function $Q(\theta^* \mid \theta)$ has been written down :

$$\begin{aligned} Q(\theta^* \mid \theta) \quad &:= \quad \mathbb{E}_\theta\big[\log L_t(\theta^*; \Delta N, X) \mid \mathbb{F}_t^{\Delta N}\big] \\ &= \quad \sum_{k=0}^{1}\sum_{i,j=1}^{n} \log D_k^*(i,j)\ \widehat{\mathcal{L}}_t^{ij,k} \end{aligned} \tag{16}$$

where $\theta^* := \{D_k^*(i,j),\ i,j = 1,\dots,n \quad k = 0,1\}$, $\mathcal{L}_t^{ij,k}$ is defined as

$$\mathcal{L}_t^{ij,k} = \sum_{l=1}^{t} 1_{\{X_{l-1}=e_i, X_l = e_j, \Delta N_l = k\}}$$

and $\widehat{\mathcal{L}}_t^{ij,k} := \mathbb{E}_\theta[\mathcal{L}_t^{ij,k} \mid \mathbb{F}_t^{\Delta N}]$ $k = 0,1$.

Maximizing the function (16) under the constraints $\sum_{k=0}^{1}\sum_{i=1}^{n}\sum_{j=1}^{n} \widehat{\mathcal{O}^{(i)}}_t D_k^*(i,j) = t$ where $\mathcal{O}_t^{(i)}$ is defined in (11), we obtain for $i,j = 1,\dots,n$

$$D_k^*(i,j) = \frac{\widehat{\mathcal{L}}_t^{ij,k}}{\widehat{\mathcal{O}^{(i)}}_t}. \tag{17}$$

As for HMCs, three different implementations of the re-estimation formula above can be carried out. We only mention the departure from the HMCs case. A common change is that each matrix $P\mathrm{diag}(G(\cdot,k))$ is replaced by D_k in the formulas.

1) Forward-backward principle: The only change is the initialization of the forward recursion for $\widehat{X}_t$ which is now $\widehat{X}_0 := x_0$ where x_0 is the probability distribution of X_0.

2) Recursive smoothing: Besides the common change in matrices notation mentioned above, Lemma 1 is valid using the new initialization of the state filter.

3) A filter-based EM algorithm: The difference equation for $\widehat{\mathcal{O}^{(i)}p_t}$ in Lemma 2 must be initialized with $\widehat{\mathcal{O}^{(i)}p_0} := x_0(i)\boldsymbol{e}_i$. The recursive form of the conditional expectation $\widehat{\mathcal{L}^{ij,k}p_t}$ is from [20]: $\widehat{\mathcal{L}^{ij,k}p_0} = 0$ and for $t \in \mathbb{N}$

$$\widehat{\mathcal{L}^{ij,k}p_{t+1}} = \frac{\widehat{\mathcal{L}^{ij,k}p_t}\ D_{\Delta N_{t+1}}}{\widehat{X}_t D_{\Delta N_{t+1}}\mathbf{1}^\top} + \frac{D_k(i,j)\widehat{X}_t(i)}{\widehat{X}_t D_k \mathbf{1}^\top} 1_{\{\Delta N_{t+1}=k\}}\ \boldsymbol{e}_j.$$

B. A continuous-time architecture-based model

A standard model in the continuous-time context was provided by Littlewood in [25]. It has inspired most other works (see [26], [6] for details). The failure process associated with such models turns to be a particular instance of the class of continuous-time Markovian Arrival Processes [24]. The well-known Poisson process modulated by a Markov chain belongs to the class of MAPs. In our context of software reliability modeling, we deal with a bivariate continuous-time Markov chain $(N,X) := (N_t, X_t)_{t\geq 0}$, where $(N_t)_{t \geq 0}$ is the counting process of failures and $(X_t)_{t\geq 0}$ is interpreted to be a Markovian model of the flow of control between the modules of a software. If the states are listed in lexicographic order, the generator of (N,X) has the form (15) where the non-negative number $D_0(i,j), j \neq i$ – $D_1(i,j)$ – represents the rate at which X jumps from state e_i to e_j with no – one – failure event. The distribution function of N_t is numerically evaluated using the uniformization technique (e.g. see [26]). As in the discrete-time case, the knowledge of the non-negative parameter vector

$$\theta = \{D_k(i,j), \quad k = 0,1 \quad i,j = 1,\dots,n\}$$

is required and we can obtain a priori estimates for θ [6]. The process (N,X) is thought of as a partially observed Markov process. The observed process is the counting process of failures and the state or hidden process is the Markov process X. The EM-algorithm is still a standard way to estimate the parameters. Specifically, it has been used by [27] for the Markov Modulated Poisson Process, by [28] for the Phase-Type distributions, by [29], [30] for general MAPs.

Here, the failure times $T_0 := 0, T_1, \dots, T_{N_t}$ and the censure data $t - T_{N_t}$ are the observations up to time t. Using the fact that the complete data consist in the set of observations with the complete path of the Markov chain X over $[0, t]$, we obtain an explicit form for the complete data likelihood function and for $Q(\theta \mid \theta^*)$ as well. Neglecting a term associated with the censure data $t - T_{N_T}$ (the term tends to 0 as the number of observations growths to infinity), the sufficient statistics for the complete data likelihood are the continuous-time counterpart of those of the discrete-time case and the re-estimation formulas for D_0, D_1 are given in (17).

1) Forward-Backward principle: All the works on parameter estimation of MAPs by EM mentioned above use the forward-backward principle to get estimates of the conditional expectations $\widehat{\mathcal{L}}_t^{ij,k}$ and $\widehat{\mathcal{O}^{(i)}}_t$ at any failure time. Assume that K values of failure times $t_1, \dots, t_K$ have been observed. The computation of the following conditional expectations is required

$$\widehat{\mathcal{O}^{(i)}}_{t_K} = \int_0^{t_K} \mathbb{P}\{X_s = e_i \mid \mathbb{F}^N_{t_K}\} \mathrm{d}s$$
$$\widehat{\mathcal{L}}^{0,ij}_{t_K} = \int_0^{t_K} \mathbb{P}\{\Delta N_s = 0, X_{s-} = e_i, X_s = e_j \mid \mathbb{F}^N_{t_K}\} \mathrm{d}s$$
$$\widehat{\mathcal{L}}^{1,ij}_{t_K} = \int_0^{t_K} \mathbb{P}\{\Delta N_s = 1, X_{s-} = e_i, X_t = e_j \mid \mathbb{F}^N_{t_K}\} \mathrm{d}s$$

where $\Delta N_t := N_t - N_{t-}$ is the increment of the counting process at time t. To go further, we introduce some additional notations

$f_0(x) := \exp(D_0 x)$ and $f_1(x) := \exp(D_0 x) D_1$
for $l = 1, \dots, K$, $\Delta t_l := t_l - t_{l-1}$ with $t_0 := 0$.

The E-step has the following form (e.g. see [30]):

Forward. $\alpha_0 := x_0$, for $l = 1, \dots, K$, $\quad \alpha_l := \alpha_{l-1} f_1(\Delta t_l)$
$c_l := \alpha_l \mathbf{1}^\top$

Backward. $\beta^\top_{K+1} := \mathbf{1}^\top$, and for $\quad l = K, \dots, 1$
$\beta_l^\top := f_1(\Delta t_l) \beta^\top_{l+1}$

For $i, j = 1, \dots, n$:
$L_0^{0,ij} := 0, L_0^{1,ij} := 0$ and for $l = 1, \dots, K$:

$$L_l^{0,ij} := L_{l-1}^{0,ij} + \alpha_{l-1} \int_{t_{l-1}}^{t_l} f_0(t - t_{l-1}) \boldsymbol{e_i}^\top D_0(i,j) \boldsymbol{e_j} f_1(t_l - t) \mathrm{d}t \ \beta^\top_{l+1}$$
$$L_l^{1,ij} := L_{l-1}^{1,ij} + \alpha_{l-1} f_0(\Delta t_l) \boldsymbol{e_i}^\top D_1(i,j) \boldsymbol{e_j} \beta^\top_{l+1}$$
$$O_l^{(i)} := O_{l-1}^{(i)} + \alpha_{l-1} \int_{t_{l-1}}^{t_l} f_0(t - t_{l-1}) \boldsymbol{e_i}^\top \boldsymbol{e_j} f_1(t_l - t) \mathrm{d}t \ \beta^\top_{l+1}.$$

The M-step is

$$\widehat{D}_1^{(m+1)}(i,j) := \frac{L_K^{0,ij}}{O_K^{(i)}}; \quad i \neq j, \ \widehat{D}_0^{(m+1)}(i,j) := \frac{L_K^{0,ij}}{O_K^{(i)}}$$

and the diagonal entries $\widehat{D}_0^{(m+1)}(i,i), i = 1, \dots, n$ are deduced from the constraints $(\widehat{D}_0^{(m+1)} + \widehat{D}_1^{(m+1)}) \mathbf{1}^\top = \mathbf{0}^\top$.

Here, the only difference with the discrete-time formulation in page 3 is that the forward quantity α_l must be normalized to 1 to get the state filter

$$\widehat{X}_{t_l} = \frac{\alpha_l}{c_l} \quad l = 1, \dots, K.$$

The conditional expectations $\widehat{\mathcal{L}}^{0,ij}_{t_K}$ and $\widehat{\mathcal{L}}^{1,ij}_{t_K}$ are given by

$$\widehat{\mathcal{O}^{(i)}}_{t_K} = \frac{O_K^{(i)}}{c_K} \quad \widehat{\mathcal{L}}^{0,ij}_{t_K} = \frac{L_K^{0,ij}}{c_K} \quad \widehat{\mathcal{L}}^{1,ij}_{t_K} = \frac{L_K^{1,ij}}{c_K}$$

2) Recursive smoothing: We do not consider the recursive smoothing for continuous-time HMCs. Indeed, some technical issues have to be overcame and some of them have to be satisfactory addressed in the context of general MAPs. We refer the interested reader to [31], [32] for the special case of a Poisson process modulated by a Markov process.

3) Filters based approach: A filter-based approach may be considered as in the discrete-time case. That is the conditional expectations $\widehat{\mathcal{O}^{(i)}}_t, \widehat{\mathcal{L}}_t^{0,ij}, \widehat{\mathcal{L}}_t^{1,ij}$ for MAPs are computed from a set of recursive equations. Such recursive equations are derived in [31] using a change of probability measure. Indeed, there exists a probability measure $\mathbb{P}_0$ under which $(N_t)_{t \geq 0}$ is the counting process of a Poisson process with intensity 1 and $(X_t)_{t \geq 0}$ is a Markov chain with generator with generator $D_0 + D_1$. Then, we have the following result.

Theorem 1: Let L_t be the likelihood ratio over the interval $[0, t]$ associated with the counting process $(N_t)_{t \geq 0}$ of intensity $\lambda_t := p_{s-} D_1 \mathbf{1}^\top$:

$$L_t := \prod_{0 < s \leq t} \lambda_s{}^{\Delta N_s} \exp\left(\int_0^t (1 - \lambda_s) \, \mathrm{d}s \right)$$

Set $\sigma(Z_t) := \mathbb{E}_0[Z_t L_t \mid \mathbb{F}_t^N]$ for any $\mathbb{F}^{N,X}$-adapted integrable process $(Z_t)_{t \geq 0}$. Let $(n_t)_{t \geq 0}$ be the process defined by $n_t := N_t - t$. We have for any $t \geq 0$

$$\sigma(p_t) = \widehat{X}_0 + \int_0^t \sigma(p_{s-}) Q \, \mathrm{d}s + \int_0^t \sigma(p_{s-})(D_1 - I) \, \mathrm{d}n_s. \tag{18a}$$

$$\begin{aligned} \sigma(\mathcal{O}_t^{(i)} p_t) = & \int_0^t \left[\sigma(\mathcal{O}_{s-}^{(i)} p_{s-}) Q + \sigma(p_{s-})(i) \, \boldsymbol{e_i} \right] \mathrm{d}s \\ & + \int_0^t \sigma(\mathcal{O}_{s-}^{(i)} p_{s-})(D_1 - I) \, \mathrm{d}n_s. \end{aligned} \tag{18b}$$

$$\begin{aligned} \sigma(\mathcal{L}_t^{0,ij} p_t) = & \int_0^t \left[\sigma(\mathcal{L}_{s-}^{0,ij} p_{s-}) Q + D_0(i,j) \sigma(p_{s-})(i) \, \boldsymbol{e_j} \right] \mathrm{d}s \\ & + \int_0^t \sigma(\mathcal{L}_{s-}^{0,ij} p_{s-})(D_1 - I) \, \mathrm{d}n_s \end{aligned} \tag{18c}$$

$$\sigma(\mathcal{L}_t^{1,ij} p_t) = \int_0^t \left[\sigma(\mathcal{L}_{s-}^{1,ij} p_{s-}) Q + D_1(i,j)\sigma(p_{s-})(i)\, \boldsymbol{e_j}\right] \mathrm{d}s$$
$$+ \int_0^t \left[\sigma(\mathcal{L}_{s-}^{1,ij} p_{s-})(D_1 - I) + D_1(i,j)\sigma(p_{s-})(i)\, \boldsymbol{e_j}\right] \mathrm{d}n_s. \tag{18d}$$

The conditional expectations under $\mathbb{P}_0$ of the statistics $\mathcal{O}_t^{(i)}, \mathcal{L}_t^{1,ij}, \mathcal{L}_t^{0,ij}$ are $\sigma(\mathcal{O}_t^{(i)}) = \sigma(\mathcal{O}_t^{(i)} p_t)\mathbf{1}^\top, \sigma(\mathcal{L}_t^{1,ij}) = \sigma(\mathcal{L}_t^{1,ij} p_t)\mathbf{1}^\top, \sigma(\mathcal{L}_t^{0,ij}) = \sigma(\mathcal{L}_t^{0,ij} p_t)\mathbf{1}^\top$. Finally, note that the conditional expectations under the original probability $\mathbb{P}$, $\widehat{\mathcal{O}^{(i)}}_t, \widehat{\mathcal{L}}_t^{0,ij}, \widehat{\mathcal{L}}_t^{1,ij}$, are obtained as follows

$$\widehat{\mathcal{O}^{(i)}}_t = \frac{\sigma(\mathcal{O}_t^{(i)})}{\sigma(1)}, \widehat{\mathcal{L}}_t^{0,ij} = \frac{\sigma(\mathcal{L}_t^{0,ij})}{\sigma(1)}, \widehat{\mathcal{L}}_t^{1,ij} = \frac{\sigma(\mathcal{L}_t^{1,ij})}{\sigma(1)}.$$

The stochastic differential equations in Theorem 1 are standard linear ode between two jumps of $(N_t)_{t\geq 0}$ or, equivalently, of $(n_t)_{t\geq 0}$. Therefore, a basic way to deal with the equations (18a-18d) is to integrate the linear ode over the interval of time between two jumps and to update the solution at the endpoint of the interval. For instance, the state filter $\sigma(p_t)$ is solution of the ode

$$\frac{\mathrm{d}}{\mathrm{d}t} q_t \quad = \quad (Q - D_1 + I) q_t = (D_0 + I) q_t$$

with initial condition $q_{t_{l-1}} := \sigma(p_{t_{l-1}})$ in the interval $[t_{l-1}, t_l[$, Then, we update the solution at time of jump t_l as follows

$$\Delta\sigma(p_{t_l}) = (D_1 - I)\sigma(p_{t_l -}) \Longrightarrow \sigma(p_{t_l}) = D_1 \sigma(p_{t_l -}).$$

In this special case, it is easily seen that (18a) has the explicit solution given by, for $t > 0$,

$$\sigma(p_t) = \exp(t)\widehat{X}_0 f_1(\Delta t_1)\cdots f_1\big(\Delta t_{N_t}\big) f_0\big(t - t_{N_t}\big)$$

and the vector of conditional probabilities $\widehat{X}_t$ is for $t > 0$

$$\widehat{X}_t = \frac{\widehat{X}_0 f_1\big(\Delta t_1\big)\cdots f_1\big(\Delta t_{N_t}\big) f_0(t - t_{N_t})}{\widehat{X}_0 f_1\big(\Delta t_1\big)\cdots f_1\big(\Delta t_{N_t}\big) f_0(t - t_{N_t})\mathbf{1}^\top}.$$

The solutions of (18a-18d) may be computed on the grid $\Pi := \{0, t_1, \ldots, t_K\}$ of the observations from the following recursive formulas : for $l = 1, \ldots, K$

$$\sigma(p_{t_l}) = \sigma(p_{t_{l-1}})\, f_1(\Delta t_l)$$
$$\sigma(\mathcal{O}_{t_l}^{(i)} p_{t_l}) = \sigma(\mathcal{O}_{t_{l-1}}^{(i)} p_{t_{l-1}})\, f_1(\Delta t_l) + \sigma(p_{t_{l-1}}) \int_{t_{l-1}}^{t_l} f_0\big(s - t_{l-1}\big)\boldsymbol{e_i}^\top \boldsymbol{e_i} f_1\big(t_l - s\big)\,\mathrm{d}s$$
$$\sigma(\mathcal{L}_{t_l}^{0,ij} p_{t_l}) = \sigma(\mathcal{L}_{t_{l-1}}^{0,ij} p_{t_{l-1}}) f_1(\Delta t_l) + \sigma(p_{t_{l-1}}) \int_{t_{l-1}}^{t_l} f_0\big(s - t_{l-1}\big)\boldsymbol{e_i}^\top D_0(i,j)\boldsymbol{e_j} f_1(t_l - s)\,\mathrm{d}s$$
$$\sigma(\mathcal{L}_{t_l}^{1,ij} p_{t_l}) = \sigma(\mathcal{L}_{t_{l-1}}^{1,ij} p_{t_{l-1}}) f_1(\Delta t_l) + \sigma(p_{t_{l-1}}) f_0(\Delta t_l)\boldsymbol{e_i}^\top D_1(i,j)\boldsymbol{e_j}.$$

The extra factor $\exp(\Delta t_l)$ is omitted in the equations above, because the estimates at a fixed instant of D_0, D_1 only require the knowledge of the conditional expectations up to a multiplicative constant. The present formulas must be compared to those generated by the forward-backward technique. The exponential matrices as well as the integral over the exponential matrices can be computed using the uniformization method.

V. Conclusion

In this paper, we discuss various principles of implementation of the EM-algorithm for missing data models. Such a class of models have been used in software reliability modeling, in particular in the architecture-based approach. Maximum likelihood estimation is carried out by this algorithm. The procedure is easily implemented and the recursive form is appealing when an "on-line" estimation method is required. Numerical experiments are reported in [11], [12], [27], [33], [30]. Though no definitive conclusion are given, they show that EM-algorithm is a robust procedure. But, the well-known drawback of the EM-algorithm is its slow convergence to the (local) solution if there exists. Finally, we mention that, for the models considered here, an example of speed-up of the convergence may be obtained using the E-step of the EM-algorithm in combination with some gradient methods for the M-step. We refer to [34], [18] for a detailed discussion on this kind of issues.

References

[1] O. Gaudoin, "Statistical tools for software reliability evaluation," Ph.D. dissertation, Universite Joseph Fourier - Grenoble I, 1990, in French.

[2] N. D. Singpurwalla and Wilson, S.P., *Statistical Methods in Software Engineering*. Springer, 1999.

[3] J. D. Musa, A. Iannino, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application.* McGraw-Hill International Editions, Computer Science Series, 1987.

[4] H. Pham, *Software Reliability*. Springer, 2000.

[5] M. R. Lyu, Ed., *Handbook of software reliability engineering*. McGraw-Hill, 1996.

[6] K. Goseva-Popstojanova and K. S. Trivedi, "Architecture-based approach to reliability assessment of software systems," *Performance Evaluation*, vol. 45, pp. 179–204, 2001.

[7] J. Ledoux, "Software reliability modeling," in *Handbook of Reliability Engineering*, H. Pham, Ed. Springer, 2003, pp. 213–234.

[8] M. Kimura and S. Yamada, "Statistical estimation of imperfect debugging rate based on hidden-markov software reliability modeling." in *5th ISSAT Int. Conf. on Reliab. and Quality in Design*, 1999, pp. 331–335.

[9] H. Okamura, Y. Watanabe, and T. Dohi, "Estimating mixed software reliability models based on the EM-algorithm." in *Int. Symp. on Empirical Software Eng. (ISESE'02).* IEEE Computer Society, 2002, pp. 69–78.

[10] H. Okamura, Y. Watanabe, and T. Dohi, "An Iterative Scheme for Maximum Likelihood Estimation in Software Reliability Modeling" in *14th IEEE Int. Symp. on Software Reliab. Eng. (ISSRE'03)*, 2003, pp. 246–256.

[11] H. Okamura, A. Murayama and T. Dohi, "EM-algorithm for discrete software reliability models : a unified parameter estimation method." in *8th IEEE Int. Symp. on High Assurance Systems Eng. (HASE'04)*, 2004, pp. 219–228.

[12] J.-B. Durand and O. Gaudoin, "Software reliability modelling and prediction with hidden Markov chains," *Statist. Modelling*, vol. 5, pp. 75–93, 2005.

[13] J. Ledoux, "Parameter reestimation for a reliability model for modular software," in *Supplementary Proc. of 15th IEEE Int. Symp. on Software Reliab. Eng. (ISSRE'04)*, StMalo, November 2004, pp. 51–52.

[14] ——, "Towards a filter-based EM-algorithm for parameter estimation of Littlewood's software reliability model," in *11th Int. Symp. on Applied Stochastic Models and Data Analysis*, 2005, pp. 1224–1231.

[15] R. J. Elliott, L. Aggoun, and J. B. Moore, *Hidden Markov Models*. Springer, 1995.

[16] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm (with discussion)." *J. Roy. Statist. Soc. Ser. B*, vol. 39, pp. 1–38, 1977.

[17] L. E. Baum, E. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Statist.*, vol. 41, pp. 164–171, 1970.

[18] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*. Springer, 2005.

[19] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications speech recognition." *Proceedings of the IEEE*, vol. 77, pp. 257–286, 1989.

[20] J. Ledoux, "Recursive filters for partially observable finite Markov chains," *J. Appl. Probab.*, vol. 42, no. 3, pp. 687–697, 2005.

[21] R. C. Cheung, "A user-oriented software reliability model." *IEEE Trans. Software Eng.*, vol. 6, pp. 118–125, 1980.

[22] J. Ledoux and G. Rubino, "Simple formulae for counting processes in reliability models," *Adv. in Appl. Probab.*, vol. 29, pp. 1018–1038, 1997.

[23] K. Goseva-Popstojanova, A. Hassan, A. Guedem, W. Abdelmoez, D. E. M. Nassar, H. Ammar, and A. Mili, "Architectural level risk analysis using UML," *IEEE Trans. Software Eng.*, vol. 29, pp. 946–960, 2003.

[24] G. Latouche and V. Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling*, ser. series on Statistics and Applied Probability. SIAM, 1999.

[25] B. Littlewood, "A reliability model for systems with Markov structure." *J. Roy. Statist. Soc. Ser. C*, vol. 24, pp. 172–177, 1975.

[26] J. Ledoux, "Availability modeling of modular software." *IEEE Trans. Reliab.*, vol. 48, pp. 159–168, 1999.

[27] T. Rydén, "An EM-algorithm for estimation in Markov-modulated Poisson process," *Computat. Statist. Data Analysis*, vol. 21, pp. 431–447, 1996.

[28] S. Asmussen, "Matrix-analytic models and their analysis," *Scand. J. Statist.*, vol. 27, pp. 193–226, 2000.

[29] L. Breuer, "An EM algorithm for batch arrival processes and its comparison to a simpler estimation procedure." *Ann. Oper. Res.*, vol. 112, pp. 123–138, 2002.

[30] A. Klemmm, C. Lindemann, and M. Lohmann, "Modeling IP traffic using the Batch Markovian Arrival Process," *Performance Evaluation*, vol. 54, pp. 149–173, 2003.

[31] J. Ledoux, "Filtering and the EM-algorithm for the Markovian Arrival Process," 2005, revised for publication.

[32] R. J. Elliott and W. P. Malcolm, "General smoothing formulas for Markov-modulated poisson observations," *IEEE Trans. Automat. Control*, vol. 50, no. 8, pp. 1123–1134, 2005.

[33] A. Lang and J. Arthur, "Parameter estimation for phase-type distributions," in *Matrix Analytic methods in Stochastic Models.*, S. Chakravarty and A. Alfa, Eds. Marcel Dekker, 1996, pp. 151–206.

[34] M. Jamshidian and R. I. Jennrich, "Accelaration of the em algorithm by using quasi-Newton methods." *J. Roy. Statist. Soc. Ser. B*, vol. 59, pp. 569–587, 1997.